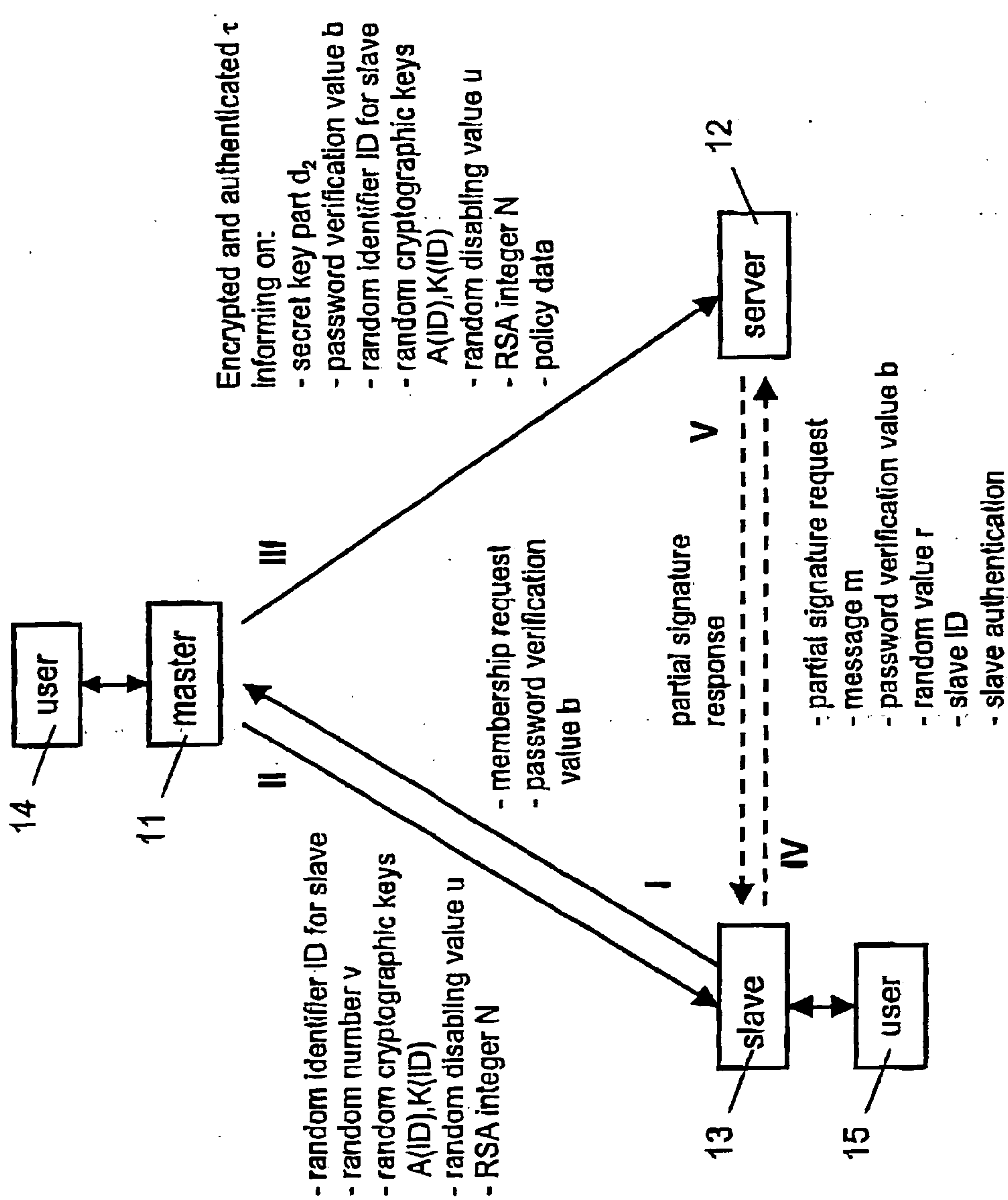


Encrypted and authenticated τ informing on:
- secret key part d_2
- password verification value b
- random identifier ID for slave
- random cryptographic keys A(ID),K(ID)
- random disabling value u
- RSA integer N
- policy data
user
master
server
slave
user
14
11
12
13
15
I
II
III
IV
V
- membership request
- password verification value b
partial signature response
- partial signature request
- message m
- password verification value b
- random value r
- slave ID
- slave authentication
- random identifier ID for slave
- random number v
- random cryptographic keys A(ID),K(ID)
- random disabling value u
- RSA integer N

FIG. 1

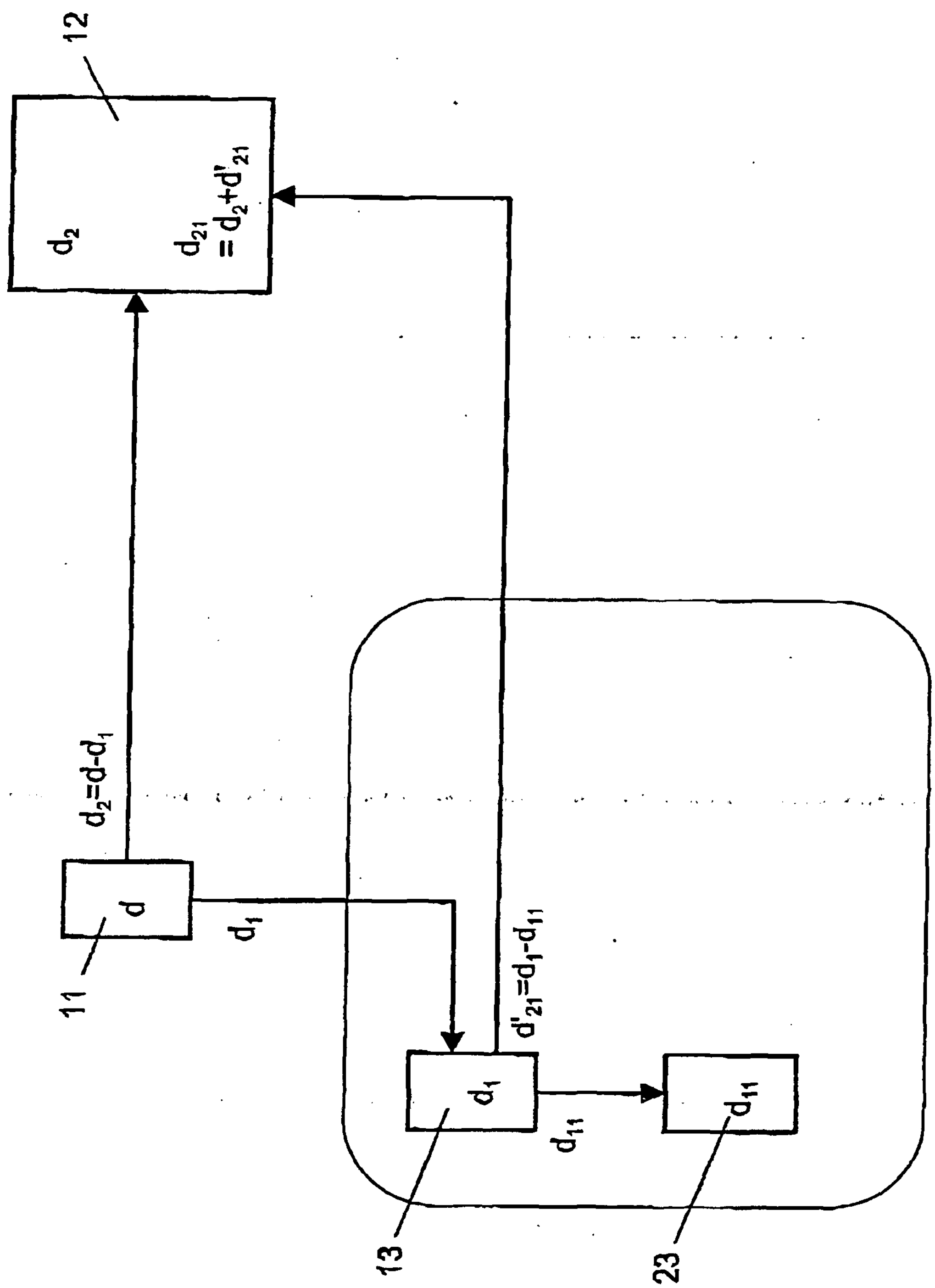


FIG. 2

A b s t r a c t

The invention relates to a method for sharing the authorization to use specific resources among multiple devices, which resources are accessible via messages on which a secret key operation was applied with a predetermined secret master key d available at a master device 11. In order to provide an optimized sharing of authorization, it is proposed that the master device 11 splits the secret master key d into two parts d_1, d_2. A piece of information relating to the first part d_1 of the secret master key d is forwarded to the slave device 13 for enabling this slave device to perform a partial secret key operation on a message m. The second part d_2 of the secret master key d is forwarded to a server 12 for enabling the server 12 to perform partial secret key operations on a message m received from the slave device 13.

For publication: Figure 1

wish to allow another person to access the account for a limited time at least to a limited extent.

A general approach for enabling a sharing of authorization is to define an authorization domain consisting of several personal devices. The authorization for a service is then granted to the domain, rather than to a specific device. A device is allowed access the service if its membership in the authorization domain can be verified.

A more specific approach for enabling the use of resources from several devices has been proposed by the IETF sacred working group in "http://www.ietf.org/html.charters/sacred-charter.html".

The IETF proposal aims at allowing users to utilize different user devices from which their authorizations can be used. To this end, two approaches are presented.

In the first approach, a user is enabled to create his/her credentials on one device and to securely upload them to a credential server. Thereafter, the user may download these credentials from the credential server to any device and use them there. The download process is controlled by an authentication of the user to the credential server. The authentication can be based in particular on passwords, since the user is not required to possess any personal device.

This first approach has the disadvantage that the credential server is an attractive point for attack. Further, depending on the details of the protocol, the

credential server itself may have to be trusted to a high degree. For example, if the credentials are stored on the server encrypted with the user's password, the server will be able to mount a dictionary attack to recover the credentials. Moreover, in order to share the same resources among different users, the user to whom the credentials belong has either to enter his/her password personally to the device of another user, which is usually not possible, or to impart the password, which is usually not desired, since the password might be used also for other applications.

In the second approach presented by IETF, credentials are transferred directly from one user device to another user device. This approach has the disadvantage that it implies that a complete transfer of the credentials from one device to another is performed. That is, after the transfer, the credentials will not be usable in the original device any more. This prevents concurrent sharing of authorizations.

In both approaches, the devices receiving the credentials also have to be trusted to a large extent, since they receive the credentials in plaintext. There is no transparent way to control what a client could do with the credentials, and it is not possible to revoke the authorizations granted to a client device. Thus, a partial sharing of authorization is not possible.

SUMMARY OF THE INVENTION

It is an object of the invention to provide an improved method for sharing the authorization to use specific resources among multiple devices.

It is in particular an object of the invention to enable a use of the same authorization concurrently from more than one device.

At the same time, the required level of trust on a server supporting the sharing of authorization is to be kept minimal. That is, the server by itself should not be able to use the shared authorization.

These objects are reached according to the invention with a method for sharing the authorization to use specific resources among multiple devices, which resources are accessible via messages on which a secret key operation was applied with a predetermined secret master key available at a master device. In the proposed method, the master device, which acts as a delegator of the authorization, splits in a first step the secret master key into a first part and a second part. The master device then forwards a piece of information relating to the first part of the secret master key to a slave device acting as a delegatee of the authorization. This piece of information enables the slave device to perform a partial secret key operation on messages based on the first part of the secret master key. Moreover, the master device forwards the second part of the secret master key to a server for enabling the server to perform a partial secret key operation on messages received from the slave device based on the second part of the secret master key.

The invention proceeds from the method presented in the above cited document by MacKenzie and Reiter. It is assumed that there is a master device that has a master secret key, e.g. a private key of a RSA key pair consisting of a private key and a public key. The master device acting alone will be able to fully utilize the authorizations granted to the public key by itself. But the master device is typically not expected to be used in day-to-day transactions. Instead, the master device delegates its authorizations fully or partially to one or more slave devices. These slave devices constitute an authorization domain. There is a network assistant (server) that helps slave devices to exercise the delegated authorization. Whenever a slave device is to be added to the authorization domain, the master device splits the available secret master key into two parts. The master device then transmits information on one part of the secret master key to the slave device and the other part directly or indirectly to the server.

With the presented method, a sharing of authorization is initialized. Now the slave device can transmit a request to the server that a partial secret key operation is to be performed on a message, and as a result the server returns a processed message, i.e. a partially signed or decrypted message. The slave is then able to compute the entire signed or decrypted message by combining the received message with a message on which the slave device applied its own part of the secret key in a partial secret key operation. Neither the server nor the slave device is able to obtain a signed or decrypted message when acting alone.

Compared to the above cited document by MacKenzie and Reiter and to the above mentioned second approach by the IETF, it is an advantage of the invention that resources may be used via several devices and by different users concurrently. The invention does not technically restrict the number of devices that can be members of the authorization domain.

Compared to the above mentioned first approach by the IETF, it is an advantage of the invention that the server acting alone cannot use the secret key.

It has to be noted that the server and the master functionalities can be placed physically into one device.

Preferred embodiments of the invention become apparent from the dependent claims.

In a preferred embodiment of the invention, a chained delegation of the authorization to access specific resources is enabled. That means that a slave device to which the authorization has been provided is able to further delegate the authorization to other slave devices. The rationale for such a feature is that even when the master device is currently unavailable, e.g. broken or lost, the user is able to expand his authorization domain as long as there is at least one slave device left to which the authorization was already delegated. Basically, the delegation between slave devices may take place in the same way as from the master device to a slave device. Since the slave device is not in possession of the entire secret master key, however,

the server adds the part of the of the secret master key available at the server for the respective delegating slave device to a received part of the partial secret key available at the delegating slave device.

In either case, the server should verify the identity of a device requesting a partial secret key operation, e.g. based on an authentication key, and of the user using the requesting device, e.g. based on an entered password, before transmitting a message on which a partial secret key operation was applied to the requesting device.

In a further preferred embodiment of the invention, the key splitting performed by a delegator is made dependent on a randomized password provided by the delegatee. It is proposed more specifically, that the delegatee generates a password verification value based on a password input by a user of the delegatee and on a first random number. This password verification value is provided to the delegator. The delegator then determines the respective first part of the secret master key based on the received password verification value and on a second random number. The piece of information which is forwarded by the delegator to the delegatee may comprise in this case the second random number. The delegatee is thereby enabled to compute the respective first part of the secret master key whenever required based on the correct password entered by the user, on the first random number used for generating the password verification value and on the received second random number.

It is an advantage of this embodiment of the invention that the necessity is avoided that users have to reveal

their long-term secrets to other users or to transfer them from one device to another, while it is at the same time ensured that only authorized users can access specific resources. The user of the device which requests an introduction into the authorization domain can choose a new password or use an old password based on which the secret master key is to be split, since the password itself is never revealed to the server or to the device from which an introduction to the authorization domain is requested. It is further an advantage that the respective first part of the secret master key does not have to be stored itself at the delegatee.

Advantageously, the master device and the server share a security association. This is an important feature, because otherwise a slave device can masquerade as the server and obtain both halves of the secret key. The security association between a master device and a server may consist of an authentication key associated to the master device, a confidentiality key associated to the master device and the lifetimes of these keys. The authentication key can be in particular a key of a symmetric authentication algorithm or a public digital signature algorithm, and the confidentiality key can be in particular a key of either a symmetric or an asymmetric algorithm. Preferably, both keys are keys of symmetric algorithms, since this increases the protocol speed and decreases the size of the message.

Further advantageously, also a security association between the respective slave device and the server is established. If this security association is based as well on symmetric mechanisms, the computation workload on

the server side is decreased, and moreover, the slave device and the master device may now share exactly the same types of security associations. This allows to extend the capability of the proposed authorization delegation to the slave device in a particular simple way.

Moreover, a confidential channel between a respective delegator and a respective delegatee should be provided, for example, by PKI (Public key infrastructure), shared keys, a physical connection, etc. This ensures that only an authorized device can be the delegatee and receive the secret information sent by the delegator. In particular, it prevents a server from masquerading as a delegatee, and obtain both halves of the secret key.

For realizing the invention, the steps of the proposed method associated to a delegating device are implemented in a delegator, i.e. in a master device and possibly in addition in one or more slave devices. The steps of the proposed method associated to a delegating device are implemented in a delegatee, i.e. in one or more slave devices. The steps of the proposed method associated to a server are implemented in a server, in particular in a network server.

Delegator and delegatee can be any electronic device that is suited to establish a communication with other electronic devices and with a server, e.g. mobile phones, PDAs, PCs, etc.

BRIEF DESCRIPTION OF THE FIGURES

Other objects, features and advantages of the present invention will become apparent from the following detailed description considered in conjunction with the accompanying drawings.

Fig. 1 illustrates a basic delegation of authorization in an embodiment of the method according to the invention; and

Fig. 2 illustrates a chained delegation of authorization in the embodiment of figure 1.

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 illustrates the delegation of an authorization in an embodiment of the method according to the invention. The figure presents to this end a master device 11, a network server 12 a slave device 13 between which messages are transmitted. To the master device 11 and the slave device 13, a respective user 14, 15 is associated.

The master device 11 is in possession of a secret key d which can be used as secret RSA exponent for signing messages in order to obtain access to specific resources, e.g. to a bank account, or to decrypt messages encrypted using the corresponding RSA public key. The authorization to make use of the secret key d at least to some extent is to be delegated to the slave device 13 by introducing the slave device 13 into an authorization domain.

It is assumed that a security association between the master device 11 and the server 12, has been established.

This may be done as part of an enrolling procedure with the server. The details of how the security association is set up is out of scope for this invention. This security association, which enables a secure transmission of data between the master device 11 and the server 12, consists of an authentication key A(master), a confidentiality key K(master) and the lifetimes of these keys. Both keys, A(master) and K(master), are keys of symmetric algorithms.

The messages transmitted between master device 11, server 12 and slave device 13 belong to a master-slave delegation protocol and are indicated in figure 1 by arrows I-V. Messages I, II and III represented by arrows with solid lines are employed for delegating an authorization from the master device 11 to the slave device 13, while messages IV and V represented by arrows with dashed lines are employed for using a delegated authorization.

In order to obtain a membership in an authorization domain, the slave device 13 first requests the user 15 to enter a password π_0 and generates a random number t'. The slave device 13 then computes a password verification value b by applying a function g on values t' and π_0, i.e. $b=g(t',\pi_0)$. The applied function g is a keyed hash function, for example HMAC-SHA1. Next the slave device 13 transmits a membership request along with value b to the master device 11. Due to random value t', the password verification value b reveals no information about the password π_0 to the master device 11. This allows the user

15 of the slave device 13 to use the same long-term password π_0 for other purposes, too.

Upon receipt of the membership request, the master device 11 asks its user 14 whether the request is to be granted. The user 14 can consent to the request by entering a valid password.

In case the user 14 consents to the request, the master device 11 then generates an identity value ID by which the server 12 can identify a specific security association that will be established between the server 12 and the requesting slave device 13. The master device 11 further generates a random authentication key A(ID) and a random confidentiality key K(ID). Keys A(ID) and K(ID) form the cryptographic parameters of the security association that will be shared between the slave device 13 and the server 12.

The master device moreover generates a random number v. The master device 11 then computes a first half-key d_1 by using generated random number v and received random number b as variables in a keyed hash function f, i.e. $d_1=f(v,b)$. By using the random number v in addition to received random number b for calculating first half-key d_1, the master device 11 does not have to trust the pseudorandom generator of the slave device 13. The master device 11 further calculates a second half-key d_2 as the difference between the available key d and the computed first half-key d_1, i.e. $d_2=d-d_1$. Finally, the master device 11 generates a disabling key u. The disabling key u can be generated for example by applying a cryptographic hash

function on some random number t. If t is sent to the server 12, it will mark the half-key d_2 as revoked.

Next, the private values that are intended for the server 12 are encrypted at the master device 11 by the key K(master) to form a token δ. The included values comprise slave authentication key A(ID), slave confidentiality key K(ID), password verification value b, disabling key u, second half-key d_2 and RSA modulus N.

Based on token δ, a dedicated membership ticket τ for slave device 13 is created. The membership ticket τ is generated by authenticating the generated ID value, token δ and, optionally, policy data with the authentication key A(master).

The optional policy data has a structure comprising, for example, a delegation bound and a content bound. The delegation bound indicates the maximum number of allowed further delegations from the slave device 13 to other slave devices, as will be explained further below. The content bound, on the other hand, is used if the message to be signed or the encrypted message contains some pre-defined structure including attributes whose values can be compared against this bound. One example of usage of this bound is fixing the allowed amount of a transaction.

From the generated values, the values v, u, ID, A(ID) and K(ID) are now transmitted from the master device 11 to the slave device 13 in message II. Message II is transmitted via a confidential channel to the slave device 13, since it contains secret keys A(ID) and K(ID).

The confidential channel can be given by a physically secure connection or be based on a cryptographic security association between the master and the slave. This security association can be based on symmetric key algorithms or public key algorithms. When setting up such security associations users may perform the initial authentication of the devices using approaches described in the documents "Enhancements to Bluetooth baseband security", in Proceedings of Nordsec 2001, Copenhagen, November 2001, by C. Gehrmann and K. Nyberg, or "The personal CA - PKI for a Personal Area Network", IST Mobile & Wireless Telecommunications Summit, Greece June 2002, by C. Gehrmann, K. Nyberg, and J. Mitchell. In case the security association is based on public key algorithms, the confidential channel is formed by encrypting message II using a public key belonging to the slave device 13. The public key can be transmitted to the master device 11 for example in message I. The master device 11 must verify the authenticity of this public key before using it. In order to enable such a verification, methods described in the above mentioned two papers can be used. For a more straightforward approach, the slave device 13 may send message I including the public key and show a fingerprint of its public key on its display. The master device 11 then shows the fingerprint of the received public key on its display. Now the user(s) 14, 15 of the devices 11, 13 can check whether the two fingerprints match. If they do, the master device 11 is authorized to proceed with the delegation transaction. A user-friendly technique for displaying public key fingerprints is to use visual hashes.

The slave device 13 stores all values received in message II and the internally generated random value t' to some secure persistent storage. Internally generated value b, in contrast, is deleted. The received and stored value v allows the slave device 13 to compute half-key d_1 with a keyed has function $f(v,\beta)$ corresponding to the keyed hash function $f(v,b)$ used by the master device 11 for computing half-key d_1. A password verification value β is calculated anew to this end each time it is required from a password π supplied by user 15 and from random number t' stored in the device 13.

With another message III transmitted from the master device 11 to the server 12, the required security association between the slave device 13 and the server 12 is established and the second half-key d_2 provided to the server 12. Message III comprises to this end the generated ticket τ, which the server 12 verifies and stores into its database. Message III can be transmitted by the master device 11 before or after the transmission of message II.

Based on the values transmitted in messages II and III, the slave device 13 is now able to perform private key operations on messages independently of the master device 11, in order to obtain access to specific resources associated to the public key of the master device 11.

The usage of such a RSA private key operation will now be explained with reference to the fourth and a fifth message IV, V indicated in figure 1.

At the beginning of the private key operation, the user 15 of the slave device 13 is requested to enter a password π, and the slave device determines a password verification value β by applying the hash function $g(t', \pi)$ on stored random number t' and received password π.

The slave device 13 then determines a string γ containing the identification value ID, a label "priv_key_op" and an encryption of the message m on which the private key operation is to be performed, of an encoding value r and of password verification value β. The encryption is performed using confidentiality key K(ID). The label "priv_key_op" indicates that the server 12 is to perform a private key operation as opposed to a further delegation operation, which will be explained further below. Next, the slave device applies the authentication algorithm using key A(ID) on the determined value γ, resulting in a value δ.

The slave device 13 then sends a partial private key operation request comprising the values γ and δ as message IV to the server 12.

When the server 12 receives values γ and δ, it will search for the ID number associated to the slave device 13 in its database. Based on the ID number, the server obtains all the information that was transmitted within τ received from the master device for this specific slave device 13, i.e. the values A(ID), b, u, d2, N and K(ID). Any further operation is aborted, in case the second half-key d_2 is disabled by disabling value u.

Subsequently, the server 12 authenticates the slave device 13. To this end, the server 12 applies the authentication algorithm using key A(ID) to the received value γ and compares the result with received value δ. In case the compared values are not equal, the procedure is aborted.

The server may then decrypt the encrypted part of γ by means of the confidentiality value K(ID), in order to obtain message m, encoding value r and password verification value β. Based on the obtained value β, the server 12 now authenticates the user 15 by verifying that β is equal to b, i.e. that the user 15 of the slave device 13 entered the correct password π. If the server 12 can authenticate the slave device 13 but not the user 15, the server 12 may keep count of successive incorrect password attempts. If the count exceeds a given bound, the server 12 may assume that the slave device 13 has been stolen and abort the procedure.

In case policy data with a content bound was comprised in the ticket τ provided to the server 12 for this slave device 13, the server 12 also checks whether the values in the message m are within the limits provided for these values by the policy data. In case the values in message m are not within these limits, the procedure is aborted.

After a successful authentication procedures, the server 12 performs a partial private key operation on the received message m and the received encoding r based on

the second half-key d_2 according to the formula $v = \text{encode}(m, r)^{d_2} (\text{mod} N)$.

Since only the original master device 11 has access to the entire private key d, it cannot be assumed that slave devices 13 acting as delegators in a chained delegation, which will be described below, could perform computations modulo $\phi(N)$. Therefore, reduction modulo $\phi(N)$ proposed in the above cited document by MacKenzie and Reiter for computing the second half-key d_2 by the master device 11 was omitted in the presented embodiment of the invention. Since d_1 is generated as an output from a hash function, it may happen that d_2 is a negative integer. If this is the case, the server 12 computes first the private key operation with the positive integer $-d_2$, and subsequently computes the inverse of the resulting number modulo N. With this convention, the server 12 can always perform partial private key operation, even if its exponent is a negative number.

Value v resulting in the partial private key operation is encrypted based on confidentiality key K(ID) and provided to the slave device 13 as encrypted value μ in message V.

When the slave device 13 receives the partial private key operation response from the server 12, it decrypts the received value μ with its confidentiality key K(ID). Further, it generates the first half-key d_1 using the stored value v and the recently generated value β by applying the above mentioned function $f(v, \beta)$.

The slave device applies the obtained half key d_1 on the message m and combines it by multiplication with the result v of the partial private key operation received from the server according to the formula $s=v \cdot encode(m,r)^{d_1}(mod N)$. The result of this computation is the desired result s, if $s^e \equiv encode(m,r)(mod\ N)$. This provides also an implicit authentication of the server 12. In case the last verification is positive, the slave device 13 may transmit the values s and r to the server providing the desired resources.

The protocol described with reference to figure 1 allows the master device 11 to delegate its rights to a slave device 13, which slave device 13 is thereby introduced into the authorization domain. There is no technical limitation on the number of slave devices that the master device 11 may introduce in this way into the authorization domain.

In the presented embodiment of the invention, also a slave device 13 which is a member of the authorization domain may introduce other slave devices into the authorization domain. This aspect of the embodiment of the invention will now be described with reference to figure 2.

In figure 2, master device 11, server 12 and slave device 13 of figure 1 are depicted again. In addition, a second slave device 23 is shown.

Based on the initialization procedure described with reference to figure 1, the first slave device 13 is able

to calculate half-key d_1, while the server 12 is in possession of a complementary half-key d_2.

The first slave device 13 is allowed to further delegate the received authorization to the second slave device 23 without having to involve the master device 11, unless the master device 11 transmitted policy data to the server 12 indicating that a further delegation is not allowed.

The procedure for the chained delegation corresponds basically to the procedure explained with reference to figure 1, except that the first slave device 13 takes the role of the master device 11. Therefore, only the differences in the processing will be described in detail. A difference is due to the fact that the first slave device 13 is only able to calculate half-key d_1, thus it is not in possession of the entire secret key d like the master device 11. Further, the first slave device 13 has to be allowed to further delegate the authorization.

Upon a delegation request by the second slave device 23 with a message corresponding to message I of figure 1, the first slave device 13 generates a further first half-key d_{11} based on a random number and provides this random number to the second slave device 23 in a message corresponding to message II of figure 1. Moreover, the first slave devices 13 calculates a value d'_{21} with $d'_{21}=d_1 - d_{11}$ and transmits it in a message corresponding to message III of figure 1 to the server 12. Next, the server 12 checks the number of delegations already made by the first slave device 13 and compares this number to

the delegation bound which was received before as policy data from the master device 11. If this number exceeds the delegation bound, then the server 12 does not allow the delegation.

In case the delegation is allowed, the server 12 adds the stored value of first half-key d_2 to the newly received value d'_{21} to obtain a value d_{21} as further second half-key. Obviously, the resulting further second half-key d_{21} is $d_{21} = d_2 + d'_{21} = d_2 + d_1 - d_{11} = d - d_{11}$. Thereby, the second slave device 23 becomes a member of the authorization domain, because the second slave device 23 and the server 12 possess half-keys d_{11}, d_{21} which allow them to share the RSA private key function. A private key operation is performed exactly as with messages IV and V explained above, where values d_1 and d_2 are substituted by values d_{11} and d_{21}.

As becomes apparent, the described embodiment of the invention maintains the advantages of the method presented by MacKenzie and Reiter in the above cited document. As in the solution of this document, the presented method according to the invention involves a minimal invasiveness, since it does not require an agreement from communication partners. Communication partners are not aware that a signature was constructed or that an encrypted message will be decrypted using the assistance of a network server. As in the solution by MacKenzie and Reiter, a minimal trust on the network servers is required, since the server by itself cannot use the private key. It only has to be trusted that the server will stop co-operating with a slave device if the disabling key for that slave device is disclosed and that

the server obeys the requested policies. As in the solution by MacKenzie and Reiter, the server verifies both, the user and the device, before the device is allowed to use an authorization.

In addition, the described method according to the invention does not put any technical restrictions on the number of devices that may become members of the authorization domain. In particular, a chained delegation between slave devices is enabled. The chained delegation does not require the availability of the master device. Still, the master device can restrict the usage of its secret key by providing appropriate policy data to the server. Each delegating party can add its own policies indicating whether it does or does not want to provide further delegation rights. The user of the respective delegatee can moreover choose a new password, or use an old password. The password itself is never revealed to the respective delegator or to the server.

It is to be noted that the described embodiment constitutes only one of a variety of possible embodiments of the invention, and also the described embodiment can be varied in many ways. A selection of possible variations will be presented in the following.

In the described embodiment of the invention, secret key d is split by the master device into half keys of equal size. In contrast to this approach, the workload of either the server or the slave device could be minimized by making its half-key particularly small, e.g. 1/10th of the size of the original key.

In the described embodiment of the invention, the master device chooses the values ID, u, A(ID) and K(ID). Alternatively, these values could be chosen as well by the server or by the slave device. If the server chooses these values, the protocol has to be interactive, i.e. the server must participate in the delegation process because these values have to be provided to the master device before message II. However, in case the master device chooses these values by itself as proposed, it does not have to rely on the quality of randomness available to the other entities.

In the described embodiment of the invention, the policy data is included directly in the membership ticket τ, i.e. without encryption. In case the policy data should remain confidential, it is also possible to include it in the data that is encrypted to token δ.

In the described embodiment of the invention, the membership ticket τ is provided directly from the respective delegator to the server. In an alternative approach, the membership ticket τ could also be provided to the server via the respective delegatee. In figure 1, for example, the membership ticket τ generated by the master device 11 could be transmitted to the slave device 13 in message II. The slave device 13 then forwards the membership ticket τ to the server 12 in message IV. In case τ is provided online, i.e. together with a request for a partial private key operation, the server must verify and decrypt τ every time when the slave device requests a partial private key operation. This can be

avoided by storing the membership ticket τ in the server the first time the slave device transmits such a request to the server. Thereafter, τ does not have to be provided again.

In another alternative, the membership ticket τ could be provided from the respective delegator directly to the server each time the respective delegatee requests a partial private key operation from the server, i.e. not in an initializing step as in the above described embodiment of the invention.

In any case, the generation of the ticket τ is separated from the use of the ticket τ.

In the above cited document by MacKenzie and Reiter, a random string ρ is employed, which is used as a one time pad for encrypting the result of the partial private key operation before it is sent from the server to the device. In the above described embodiment of the invention, instead an encryption of the result v with a confidentiality key K(ID) is employed. This is not necessary. The computational workload of the server can be further reduced, if the slave provides the server with such a one time pad ρ encrypted as part of the string γ in message IV to be used by the server to encrypt its reply message V to the slave device.

Claims

1. Method for sharing the authorization to use specific resources among multiple devices (11,13), which resources are accessible via messages on which a secret key operation was applied with a predetermined secret master key (d) available at a master device (11), said method comprising:
   - splitting said secret master key (d) at said master device (11) into a first part (d_1) and a second part (d_2), wherein said master device (11) is acting as a delegator of said authorization;
   - forwarding a piece of information to a slave device (13) acting as a delegatee of said authorization, which piece of information enables said slave device (13) to perform a partial secret key operation on messages (m) based on said first part (d_1) of said secret master key (d); and
   - forwarding said second part (d_2) of said secret master key (d) to a server (12) for enabling said server (12) to perform a partial secret key operation on messages (m) received from said slave device (13) based on said second part (d_2) of said secret master key (d).

2. Method according to claim 1, wherein a delegatee (13) to which said authorization was delegated is enabled to act as delegator for delegating said authorization to another slave devices (23) acting as delegatee, said method comprising for said further delegation:

- splitting a first part (d_1) of said secret master key (d) which can be generated at said delegator (13) into a further first part (d_{11}) of said secret master key (d) and another part (d'_{21});
- forwarding a piece of information to said delegatee (23), which piece of information enables said delegatee (23) to perform a partial secret key operation on messages (m) based on said further first part (d_{11});
- forwarding said other part (d'_{21}) of said first part (d_1) of said secret master key (d) to said server (12); and
- combining a second part (d_2) of said secret master key (d) available at said server (12) for said delegator (13) with said other part (d'_{21}) provided by said delegator (13) to a further second part (d_{21}) of said secret master key (d) for enabling said server (12) to perform a partial secret key operation on messages (m) received from said delegatee (23) based on said further second part (d_{21}) of said secret master key (d).

3. Method according to claim 1 or 2, wherein said step of splitting a key (d, d_1) at a respective delegator (11,13) into two parts is preceded by the steps of
   - generating a password verification value (b) at a respective delegatee (13,23) based on a password input by a user (15) of said delegatee (13,23) and on a first random number; and
   - providing said password verification value (b) to said delegator (11,13);

   wherein said respective first part (d_1, d_{11}) of said secret master key (d) is determined at said delegator

(11,13) based on said password verification value (b) received from said delegatee (13,23) and on a second random number (v) and wherein said piece of information which is forwarded by said delegator (11,13) to said delegatee (13,23) comprises said second random number (v) for enabling said delegatee (13,23) to generate said respective first part (d_1,d_{11}) of said secret master key (d).

4. Method according to one of the preceding claims, wherein said delegator (11,13) determines a respective second part (d_2,d'_{21}) of an available secret key (d,d_1) as the difference between said available secret key (d,d_1) and a randomly generated first part (d_1,d_{11}) of said secret master key (d).

5. Method according to one of the preceding claims, wherein a delegator (11,13) provides in addition policy data to said server (12) restricting the bounds of the authorization that may be delegated to a delegatee (13,23).

6. Method according to claim 5, wherein said bounds comprise a delegation bound indicating the maximum number of allowed further delegations of said authorization by a respective delegatee (13) acting as a delegator for further delegatees (23).

7. Method according to claim 5 or 6, wherein said bounds are content bounds comprising at least one value which can be compared to the values of attributes in a message (m) on which a secret key operation is to

be performed, said message (m) having a pre-defined structure including said attributes.

8. Method according to one of the preceding claims, wherein said delegator (11,13) transmits a respective second part (d_2,d'_{21}) of an available secret key (d,d_1) computed for a specific delegatee (13,23) directly to said server (12) once during an initialization process for a specific delegatee (13,23).

9. Method according to one of claims 1 to 7, wherein said delegator (11,13) transmits a respective second part (d_2,d'_{21}) of an available secret key (d,d_1) computed for a specific delegatee (13,23) directly to said server (12) upon a request by said server (12) each time said specific delegatee (13,23) requests a partial secret key operation on a message (m).

10. Method according to one of claims 1 to 7, wherein said delegator (11,13) transmits a respective second part (d_2,d'_{21}) of an available secret key (d,d_1) computed for a specific delegatee (13,23) to said server via said specific delegatee (13,23) once during an initialisation process.

11. Method according to one of claims 1 to 7, wherein said delegator (11,13) transmits a respective second part (d_2,d'_{21}) of an available secret key (d,d_1) computed for a specific delegatee (13,23) to said server (12) via said specific delegatee (13,23), said specific delegatee (13,23) forwarding said respective second part (d_2,d'_{21}) to said server (12) each time it

requests a partial secret key operation on a message (m) from said server (12).

12. Method according to one of the preceding claims, wherein a confidential channel can be established between a respective delegator (11,13) and a respective delegatee (13,23) for securely transmitting confidential information between said delegator (11,13) and said delegatee (13,23).

13. Method according to one of the preceding claims, wherein a security association is formed between a respective delegator (11,13) and said server (12) for securely transmitting confidential information between said delegator (11,13) to said server (12).

14. Method according to claim 13, wherein said security association is realized with a symmetric algorithm using cryptographic parameters (K(ID),A(ID)) associated to said delegator (11,13), which cryptographic parameters (K(ID),A(ID)) are available at said delegator (13) and at said server (12).

15. Method according to one of the preceding claims, wherein a security association is formed between a respective delegatee (13,23) and said server (12) for securely transmitting confidential information between said delegatee (13,23) and said server (12).

16. Method according to claim 15, wherein said security association is realized with a symmetric algorithm using cryptographic parameters (K(ID),A(ID))

associated to said delegatee (13) and available at said delegatee (13) and at said server (12).

17. Method according to claim 16, wherein said cryptographic parameters (K(ID),A(ID)) associated to said delegatee (13) are generated by the respective delegator (11) and provided to said delegatee (13) and to said server (12).

18. Method according to one of the preceding claims, wherein a delegatee (13,23) makes use of a delegated authorization by transmitting a request to perform a partial secret key operation on an included message (m) to said server (12), said server (12) performing a partial secret key operation on said received message (m) based on a respective second part (d_2,d_{21}) of said secret master key (d) and transmitting a resulting message as response message to said delegatee (13,23), and wherein said delegatee (13,23) performs a partial secret key operation on said transmitted message (m) based on said computed first part (d_1,d_{11}) of said secret master key (d) and combines a resulting message with said response message received from said server (12).

19. Method according to claim 18, wherein a delegator (11,13) transmits to said server (12) a password verification value (b) provided by a respective delegatee (13,23) to said delegator (11,13) during the delegation of said authorization, which password verification value (b) is generated by said delegatee (13,23) based on a password entered by a user (15) of said delegatee (13,23) and on a random number,

wherein said delegatee (13,23) transmits to said server (12) together with each request to perform a partial secret key operation on a message (m) a password verification value (β) generated by said delegatee (13,23) based on a password entered by a user (15) of said delegatee (13,23) for the respective request and on said random number, and wherein said server (12) verifies the identity of a user (15) using said delegatee (13,23) before performing said requested partial secret key operation by comparing said password verification values (b,β) received from said delegator (11,13) and from said delegatee (13,23).

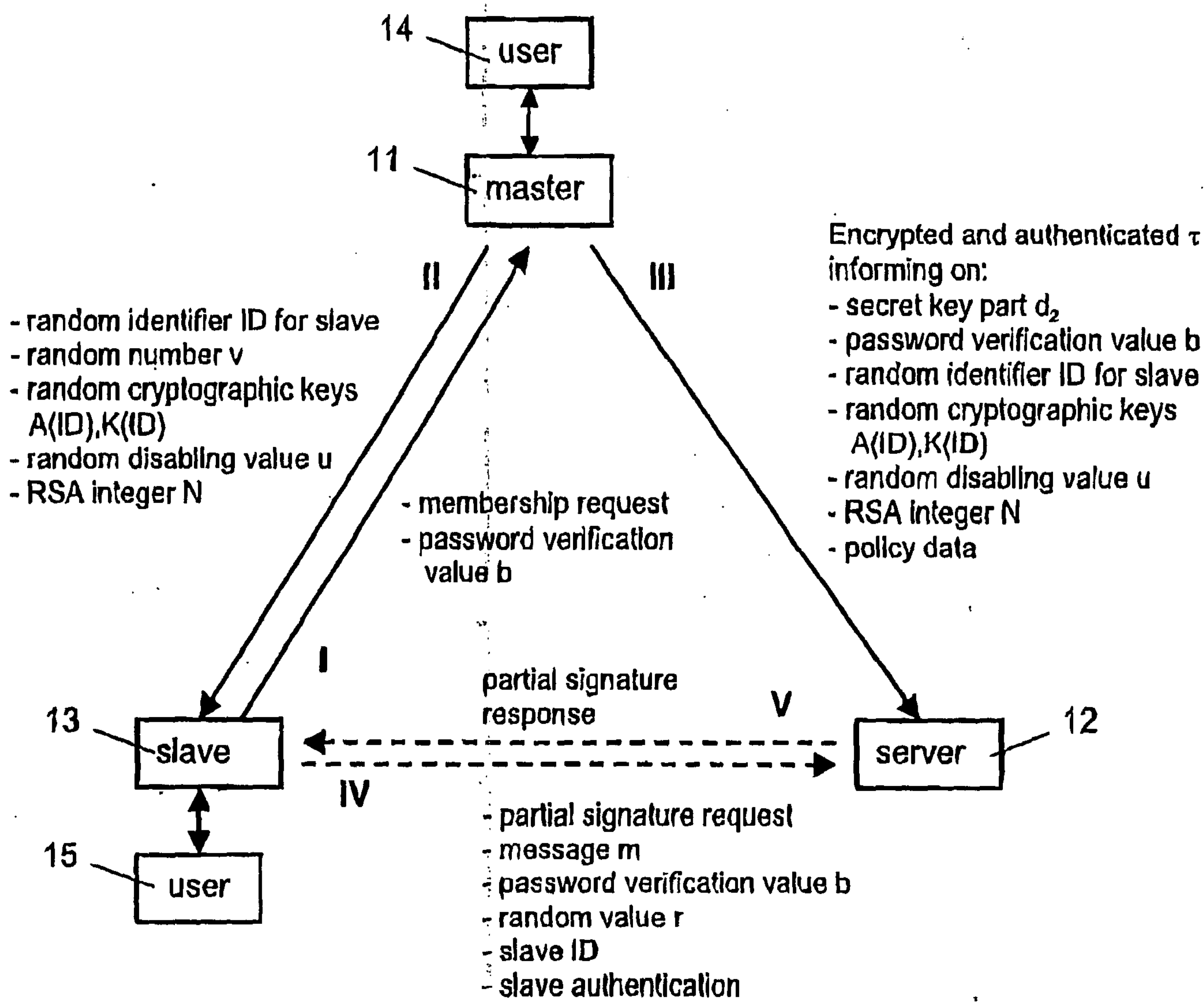
20. Method according to one of claims 18 or 19, wherein said server (12) verifies the identity of a delegatee (13,23) requesting a partial secret key operation on a message (m) before performing a requested partial secret key operation on a received message (m).

21. Delegator (11,13) comprising means for delegating an authorization to use specific resources to a delegatee (13,23) according to one of the preceding claims.

22. Delegatee (13,23) comprising means for requesting and receiving an authorization to use specific resources from a delegator (11,13) according to one of claims 1 to 20.

23. Server comprising means for supporting a chained delegation of an authorization to use specific

resources from a respective delegator (11,13) to a respective delegatee (13,23) according to one of claims 2 to 20.



14
user
11
master
II
III
- random identifier ID for slave
- random number v
- random cryptographic keys
A(ID),K(ID)
- random disabling value u
- RSA integer N
Encrypted and authenticated τ
informing on:
- secret key part d2
- password verification value b
- random identifier ID for slave
- random cryptographic keys
A(ID),K(ID)
- random disabling value u
- RSA integer N
- policy data
- membership request
- password verification
value b
I
partial signature
response
V
13
slave
server
12
IV
15
user
- partial signature request
- message m
- password verification value b
- random value r
- slave ID
- slave authentication

FIG. 1

Europäisches Patentamt
Eur pean Patent Office
Office eur péen des brevets

Bescheinigung
Certificate
Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

02015842.4

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

EPA/EPO/OEB Form 1014.1 - 02.2000 7001014

Anmeldung Nr:
Applicati n n .: 02015842.4
Demande no:

Anmeldetag:
Date of filing: 16.07.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Nokia Corporation
Keilalahdentie 4
02150 Espoo
FINLANDE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method for sharing the authorization to use specific resources

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H04L/

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR

SC/WY 020657EP
July 16, 2002

Method for sharing the authorization to use specific resources

FIELD OF THE INVENTION

The invention relates to a method for sharing the authorization to use specific resources among multiple devices, which resources are accessible via messages on which a secret key operation was applied with a predetermined secret key available at one of these devices. The invention relates equally to such devices and to a server supporting a sharing of authorization.

BACKGROUND OF THE INVENTION

It is known from the state of the art to provide an access to specific resources via a network only upon messages on which a secret key operation was performed. Such a secret key operation can be in particular signing the message digitally with a secret key or decrypting a received encrypted message based on a secret key. For example, bank account payment transactions or the purchase of rights for a piece of digital content may be enabled on-line with digitally signed messages.

Methods for generating digital signatures on messages in a distributed manner are proposed for example in the document "Networked cryptographic devices resilient to capture" in Proceedings of the 2001 IEEE Symposium on Security and Privacy, pp. 12-25, May 2001, by P. MacKenzie and M.K. Reiter. The presented methods are

aimed at minimizing the impact of stolen devices by using a network server. They are based more specifically on function sharing between a device and a network server, e.g. on sharing a secret RSA signing key. For sharing a secret RSA (Rivest, Shamir and Adelman encryption) signing key d available at a device, the device provides a half-key d_2 to an untrusted server. Whenever needed, the device can recover the complementary half-key d_1 by asking the user to enter a password. The half-keys d_1 and d_2 satisfy the relation $d=d_1+d_2 \pmod{\phi(N)}$, where $N=pq$ is the RSA modulus, where p and q are different secret prime numbers available at the device, and where $\phi(N)=(p-1)(q-1)$. After the initialization process, the secret values d, p and q will be deleted at the device. The user can then generate a signature on a message m by requesting a partial signature $m^{d2} \pmod N$ from the server. Thereafter, the device can compute the entire signature based on the generated second half-key d_1 according to the equation $m^d=m^{d1}*m^{d2} \pmod N$.

It is an underlying assumption of this method that there is only one device that uses the authorizations granted with a key pair d_1, d_2.

In some situations it might be desirable, however, to be able to use specific resources from several devices and/or by several users. An owner of a bank account which can be accessed on-line might wish to be able to access the account via several devices, for instance via a small mobile phone and a larger PDA (personal digital assistant). An owner of such a bank account might further